



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/035,604	10/26/2001	Antti Huima	35997-215548	3832

26694 7590 07/10/2007  
VENABLE LLP  
P.O. BOX 34385  
WASHINGTON, DC 20043-9998

EXAMINER

NGUYEN, THUONG

ART UNIT	PAPER NUMBER
----------	--------------

2155

MAIL DATE	DELIVERY MODE
-----------	---------------

07/10/2007

PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	<b>Application No.</b> 10/035,604	<b>Applicant(s)</b> HUIMA, ANTTI	
	<b>Examiner</b> Thuong (Tina) T. Nguyen	<b>Art Unit</b> 2155	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

- 1) ☒ Responsive to communication(s) filed on 25 June 2007.
- 2a) ☐ This action is **FINAL**.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

- 4) ☒ Claim(s) 1-3, 7-13, 16-20, 22-28 and 30 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-3, 7-13, 16-20, 22-28 & 30 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some    \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)                                | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                       | 5) <input type="checkbox"/> Notice of Informal Patent Application                       |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

### **DETAILED ACTION**

1. This action is in response to application 10/035,604 filed 6/25/07. Claims 1-3,7-13,16-20,22-28 and 30 are pending and represent method, computer program software, computer network and system for method for managing compiled filter code.

### **Claim Objections**

2. Claim 17 objected to under 37 CFR 1.75(c) as being in improper form because a multiple dependent claim 1to 3 or 7 to 11. See MPEP § 608.01(n). Accordingly, the claim 17 not been further treated on the merits. Appropriate correction required.
3. Claim 11 objected to because of the following informalities: undetermined a preamble portion of the claim. Appropriate correction is required.

### **Claim Rejections - 35 USC § 101**

4. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

5. Claims 12, 17-18, 26 are rejected under 35 U.S.C. 101 because the claimed invention of the claims 12, 17-18, 26 are directed to non-statutory subject matter.

Claims 12, 17-18, 26 recited "computer software program product for processing data packets based on compiled filter code comprising computer program code means for

managing the compiled filter code in a plurality of pieces, computer program code means: " which are adapted to perform some steps.

The computer program and the program are non-statutory as not being tangible embodied in computer readable medium in a manner so as to be executable, and also claimed that the computer program/programming execute in a computer or by a computer are descriptive material per se and are not statutory because they are not capable of causing functional change in the computer (See MPEP section 2106, Seventh Edition, Revision No. dated February 2000, at page 2100-10 and 2100-11).

Other dependent claims, which are not specifically cited above are also rejected because of the deficiencies of their respective parent claims.

### ***Claim Rejections - 35 USC § 112***

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Claims 1, 12, 17, 18 and 26 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. It's unclear to the examiner how to determine "certain point" in time? When would be the appropriate time to create and process the codes?

8. Claims 1, 12, 17, 18 and 26 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the

subject matter which applicant regards as the invention. It's unclear to the examiner what is it the applicant trying to accomplish through those steps? What is the end result of the claim limitations? Why using shadow paging for this process? Why processing packets received after and before said certain point in time? What are code pages contain?

9. Claims 11 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. It's unclear to the examiner what are  $v(x)$ ,  $r(x)$  and  $\text{mod}M$ ? Why calculating  $v(x)$ ? What is the value for  $\text{mod}M$ ?

10. Regarding claims 11, the phrase "substantially" renders the claim indefinite because it is unclear whether the limitation(s) following the phrase are part of the claimed invention. See MPEP § 2173.05(d). Appropriate correction is required.

11. Claims 11 recites the limitation "the order, the formula, the reference" in using The without mentioned in the previous part. There is insufficient antecedent basis for this limitation in the claim. Appropriate correction is required.

### **Claim Rejections - 35 USC § 102**

12. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

Art Unit: 2155

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

13. Claims 18-19, 26-27 are rejected under 35 U.S.C. 102(e) as being anticipated by

Stack Patent No. 6,257,774 B1. Stack teaches the invention as claimed including

application program and documentation generator system and method (see abstract).

14. As to claim 18, Stack teaches a computer network node for processing of data packets according to compiled filter code comprising means for managing the compiled filter code in a plurality of pieces, and

means for processing packets according to at least one first set of code pages (figure 1-4; col 3, lines 55-65; Stack discloses that the computer network node that processing the predetermined application program),

means for creating a second set of code pages to represent the set of code pages to represent the set of code pages to be used after a certain point in time (figure 5; col 5, lines 43- col 6, lines 16; Stack discloses that the computer network node that creating a set of program code),

means for processing packets received after said certain point in time according to said second set of code pages (col 10, lines 27-61; Stack discloses that the computer network node that processing a set of program code), and

means for processing packets received before said certain point in time according to said at least one first set of code pages (figure 3-4, 7; col 11, lines 50 – col

Art Unit: 2155

12, lines 50; Stack discloses that the computer network node of processing program node accordingly).

15. As to claim 19, Stack teaches a computer network node as recited in claim 18 further comprising:

means for incrementally compiling at least one rule and for producing at least one piece of code (col 7, lines 14-46; Stack discloses the computer network node of compiling the program rule begins at a logical root rule and proceeds through the ordered set of rules), and

means for updating a memory means with said at least one piece of code (col 8, lines 23-40; col 21, table II; Stack discloses the computer network node of updating the form and the program code in the non-volatile storage medium).

16. As to claim 26, Stack teaches a system for processing of data packets according to compiled filter code comprising means for managing the compiled filter code in a plurality of pieces (figure 1 & 3; col 5, lines 43 – col 6, lines 16; col 10, lines 27-61; Stack discloses the system of divided the program rules or code into many sub-programs, segments, files and records),

means for processing packets according to at least one first set of code pages (figure 1-4; col 3, lines 55-65; Stack discloses that the system that processing the predetermined application program),

means for creating a second set of code pages to represent the set of code pages to be used after a certain point in time (figure 5; col 5, lines 43- col 6, lines 16; Stack discloses that the system that creating a set of program code),

means for processing packets received after said certain point in time according to said second set of code pages (col 10, lines 27-61; Stack discloses that the system that processing a set of program code), and

means for processing packets received before said certain point in time according to said at least one first set of code pages (figure 3-4, 7; col 11, lines 50 – col 12, lines 50; Stack discloses that the system of processing program node accordingly).

17. As to claim 27, Stack teaches a system as recited in claim 26 comprising:

means for incrementally compiling a set of rules and for producing at least one piece of code (col 7, lines 14-46; Stack discloses the system of compiling the program rule begins at a logical root rule and proceeds through the ordered set of rules), and

means for updating a memory means with said at least one piece of code (col 8, lines 23-40; col 21, table II; Stack discloses the system of updating the form and the program code in the non-volatile storage medium).

### **Claim Rejections - 35 USC § 103**

18. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.



19. Claims 1-3, 7-10, 12-13, 16-17, 20, and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Stack Patent No. 6,257,774 B1 in view of Herrell, Patent No. 5,301,287.

Stack teaches the invention substantially as claimed including application program and documentation generator system and method (see abstract).

20. As to claim 1, Stack teaches a method for managing compiled filter code for processing data packets wherein compiled filter code is managed in a plurality of pieces,

processing packets according to at least one first set of code pages (figure 1-4; col 3, lines 55-65; Stack discloses that the method that processing the predetermined application program),

creating a second set of code pages to represent the set of code pages to be used after a certain point in time (figure 5; col 5, lines 43- col 6, lines 16; Stack discloses that the method that creating a set of program code),

processing packets received after said certain point in time according to said second set of code pages (col 10, lines 27-61; Stack discloses that the method that processing a set of program code), and

processing packets received before said certain point in time according to said at least one first set of code pages (figure 3-4, 7; col 11, lines 50 – col 12, lines 50; Stack discloses that the system of processing program node accordingly).

But Stack failed to teach the claim limitation wherein said pieces are pages having a predetermined length and shadow paging is used.

However, Herrell teaches user scheduled direct memory access using virtual addresses (see abstract). Herrell teaches the limitation wherein said pieces are pages having a predetermined length and shadow paging is used (col 15, lines 38-55).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Stack in view of Herrell so that the program code runs without the necessity of executing instructions in the host processor. One would be motivated to do so to started running the program without further host processor intervention, and hence data transmission is handle without the need to execute further instructions in the host processor.

21. As to claim 2, Stack and Herrell teach a method as recited in claim 1 comprising the steps of:

incrementally compiling at least one rule for obtaining a piece of code by a rule compiling entity (col 7, lines 14-46; Stack discloses the method of compiling the program rule begins at a logical root rule and proceeds through the ordered set of rules),

transmission of said piece of filter code from a rule compiling entity to a packet processing entity (figure 2 & 4; Stack discloses the method of transmission the program code or rule by the rule table, control table and process sequences),

pausing of processing of packets by said packet processing entity (col 8, lines 5-23; Stack discloses the method of processing the syntax rule that specify higher level functions or required different operations in the generation of the code or by user

Art Unit: 2155

defined; Therefore, the system would pause or run according to the user's preferences),

writing of said piece of filter code to memory means (col 8, lines 24-40; Stack discloses the method of storing the program code to the memory or storage), and

continuing of processing of packets by said packet processing entity (col 14, table I; Stack discloses the method of defining the next segment accordingly;

Therefore, the program would run or stop depend on user's preferences or setting).

22. As to claim 3, Stack and Herrell teach a method as recited in claim 1 comprising the steps of:

incrementally compiling at least one rule for obtaining a piece of code by a rule compiling entity (col 7, lines 14-46; Stack discloses the method of compiling the program rule begins at a logical root rule and proceeds through the ordered set of rules),

signaling from said rule compiling entity to a packet processing entity that a new piece of code is compiled (col 14, table I; col 21, table II; Stack discloses the method of popup the call or display according to the user's preferences or setting),

signaling from said packet processing entity to said rule compiling entity that said packet processing entity is ready for storage of said piece of code (col 14, table I; col 21, table II; Stack discloses the method of popup the call or display according to the user's preferences or setting),

writing said piece of code to a memory means (col 8, lines 24-40; Stack discloses the method of storing the program code to the memory or storage), and

signaling from said rule compiling entity to said packet processing entity that said piece of code is written to said memory means (col 16, table II; Stack discloses the method of popup the call or display according to the user's preferences or setting).

23. As to claim 7, Stack and Herrell teach a method as recited in claim 1 wherein within said step of creating a second set of code pages the steps of assigning members of an existing code pages set to be members of said second set of code pages, and removing a code page from said second set of code pages (figure 5-7; col 8, lines 64 – col 9, lines 18; col 9, lines 40-65; Stack discloses that the method of validating the member of the program rules and segments).

24. As to claim 8, Stack and Herrell teach a method as recited in claim 1 wherein within said step of creating a second set of code pages the steps of creating new code pages, and assigning said new code page to be a member of said second set of code pages (figure 5-7; col 8, lines 64 – col 9, lines 18; col 9, lines 40-65; Stack discloses that the method of creating program code and checking for the set of rules).

25. As to claim 9, Stack and Herrell teach a method as recited in claim 1 wherein removing a code page from the memory element storing the code pages, when the code page is not anymore a member of any set of code pages in use (figure 5-7; col 8, lines 64 – col 9, lines 18; col 9, lines 40-65; Stack discloses that the method of storing the program code and syntax and program rules).

26. As to claim 10, Stack and Herrell teach a method as recited in claim 4 wherein each page of code is associated with a reference number for observing the order of the

code pages (col 10, lines 16-26; col 11, lines 50 – col 12, lines 48; Stack discloses the method of running the program sequences and orderly).

27. As to claim 12, Stack teaches a computer software program product for processing data packets based on compiled filter code comprising computer program code means for managing the compiled filter code in a plurality of pieces,

computer program code means for processing packets according to at least one first one first set of code pages (figure 1-4; col 3, lines 55-65; Stack discloses that the computer software program that processing the predetermined application program),

computer program code means for creating a second set of code pages to represent the set of code pages to be used after a certain point in time (figure 5; col 5, lines 43- col 6, lines 16; Stack discloses that the computer software program that creating a set of program code),

computer program code means for processing packets received after said certain point in time according to said second set of code pages (col 10, lines 27-61; Stack discloses that the computer software program that processing a set of program code), and

computer program code means for processing packets received before said certain point in time according to said at least one first set of code pages (figure 3-4, 7; col 11, lines 50 – col 12, lines 50; Stack discloses that the computer software program of processing program node accordingly).

But Stack failed to teach the claim limitation wherein computer program code means for implementing shadow paging of pages of filter code.

However, Herrell teaches the limitation wherein computer program code means for implementing shadow paging of pages of filter code (col 15, lines 38-55).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Stack in view of Herrell so that the program code runs without the necessity of executing instructions in the host processor. One would be motivated to do so to started running the program without further host processor intervention, and hence data transmission is handle without the need to execute further instructions in the host processor.

28. As to claim 13, Stack and Herrell teach a computer software program product as recited in claim 12 further comprising:

computer program code means for incrementally compiling at least one rule and for producing at least one piece of code (col 7, lines 14-46; Stack discloses the computer software program of compiling the program rule begins at a logical root rule and proceeds through the ordered set of rules), and

computer program code means for updating a memory means with said at least one piece of code (col 8, lines 23-40; col 21, table II; Stack discloses the computer software program of updating the form and the program code in the non-volatile storage medium).

29. As to claim 16, Stack and Herrell teach a computer software program product as recited in claim 12 wherein the computer software program product is a software routine library (figure 2).

30. Claim 17 do not teach or defined any new limitation above claims 1-11 and therefore is rejected for similar reasons.

31. As to claim 20, Stack teaches a computer network node as recited in claim 18. But Stack failed to teach the claim limitation wherein means for implementing shadow paging of pages of filter code.

However, Herrell teaches the limitation wherein means for implementing shadow paging of pages of filter code (col 15, lines 38-55).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Stack in view of Herrell so that the program code runs without the necessity of executing instructions in the host processor. One would be motivated to do so to started running the program without further host processor intervention, and hence data transmission is handle without the need to execute further instructions in the host processor.

32. As to claim 28, Stack teaches a system as recited in claim 26. But Stack failed to teach the claim limitation wherein means for implementing shadow paging of pages of filter code.

However, Herrell teaches the limitation wherein means for implementing shadow paging of pages of filter code (col 15, lines 38-55).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Stack in view of Herrell so that the program code runs without the necessity of executing instructions in the host processor. One would be motivated to do so to started running the program without further host processor intervention, and hence

data transmission is handle without the need to execute further instructions in the host processor.

33. Claim 11 rejected under 35 U.S.C. 103(a) as being unpatentable over Stack, Patent No.6,257,774 B1 in view of Kyker, Patent No. 6,467,027 B1.

Stack teaches the invention substantially as claimed including application program and documentation generator system and method (see abstract).

34. As to claim 11, Stack teaches a method for managing compiled filter code for processing data packets wherein:

compiled filter code is managed in a plurality of pieces, each page of code is associated with a reference number for observing the order of the code pages (figure 1-5; col 5, lines 43 – col 6, lines 16; col 10, lines 27-61; Stack discloses the method of compiling program codes).

But Stack failed to teach the claim limitation wherein the order of any two code pages is determined by comparing values of  $v(x)$  calculated from the reference numbers associated with the code pages,  $v(x)$  being calculated substantially by the formula  $v(x) = r(x) - r(b) \text{ mod } M$  where  $r(x)$  is the reference number associated with a code page  $x$  being compared,  $r(b)$  the reference number of the base code page, and  $M$  the size of the set of allowed reference numbers  $\{0, 1, 2, \dots, M-1\}$ .

However, Kyker teaches method and system for an inuse field resource management scheme (see abstract). Kyker teaches the limitation wherein the order of



any two code pages is determined by comparing values of  $v(x)$  calculated from the reference numbers associated with the code pages,  $v(x)$  being calculated substantially by the formula  $v(x) = r(x) - r(b) \bmod M$  where  $r(x)$  is the reference number associated with a code page  $x$  being compared,  $r(b)$  the reference number of the base code page, and  $M$  the size of the set of allowed reference numbers  $\{0, 1, 2, \dots, M-1\}$  (col 6, lines 1-43).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Stack in view of Kyker so that the system would compare the reference numbers to determine the order pages. One would be motivated to do so to sort the program code pages accordingly.

35. Claims 22-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Stack, Patent No. 6,257,774 B1 in view of Wesinger, Patent No. 6,052,788.

Stack teaches the invention substantially as claimed including application program and documentation generator system and method (see abstract).

36. As to claim 22, Stack teaches a computer network node as recited in claim 18. But Stack failed to teach the claim limitation wherein the node is a virtual private network node.

However, Wesinger teaches firewall providing enhanced network security and user transparency (see abstract). Wesinger teaches the limitation wherein the node is a virtual private network node (figure 1).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Stack in view of Wesinger so that the system could process remotely. One would be motivated to do so to broaden the flexibility of the system.

37. As to claim 23, Stack teaches a computer network node as recited in claim 18. But Stack failed to teach the claim limitation wherein the node is a router node.

However, Wesinger teaches the limitation wherein the node is a router node (figure 4).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Stack in view of Wesinger so that the packet could transfer from one machine to another in the WAN network. One would be motivated to do so to connect over a wide geographical area via one or more dedicated connections.

38. As to claim 24, Stack teaches a computer network node as recited in claim 18. But Stack failed to teach the claim limitation wherein the node is a firewall node.

However, Wesinger teaches the limitation wherein the node is a firewall node (figure 1).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Stack in view of Wesinger so that the system would maintain the security for the system. One would be motivated to do so to ensured the security for the processing process.

39. As to claim 25, Stack teaches a computer network node as recited in claim 18. But Stack failed to teach the claim limitation wherein the node is a workstation.

However, Wesinger teaches the limitation wherein the node is a workstation (figure 5).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Stack in view of Wesinger so that the client could oversee what is going on between the systems. One would be motivated to do so to broaden the flexibility of the system.

40. Claim 30 is rejected under 35 U.S.C. 103(a) as being unpatentable over Stack, Patent No. 6,257,774 B1 in view of Kloth, Patent No. 6,598,034 B1.

Stack teaches the invention substantially as claimed including application program and documentation generator system and method (see abstract).

41. As to claim 30, Stack teaches a system as recited in claim 26. But Stack failed to teach the claim limitation wherein a memory component having a first access port and a second access port, and means for processing data packets, said means for processing data packets being arranged to access said memory component via said first access port, and said means for managing the compiled filter code being arranged to access said memory component via said second access port.

However, Kloth teaches rule based IP data processing (see abstract). Kloth teaches the limitation wherein a memory component having a first access port and a second access port (figure 5), and means for processing data packets, said means for processing data packets being arranged to access said memory component via said

Art Unit: 2155

first access port (figure 11), and said means for managing the compiled filter code being arranged to access said memory component via said second access port (figure 7).

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Stack in view of Kloth so that the system could processing the IP packets by dividing the packet into plurality pieces and process the packet one at a time by the communication protocol stack and the communications rule editor. One would be motivated to do so to broaden the flexibility of the system and applied the specific rule to specific packet to fasten the processing process.

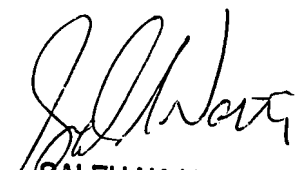
#### **Contact Information**

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Thuong (Tina) Nguyen whose telephone number is 571-272-3864, and the fax number is 571-273-3864. The examiner can normally be reached on 8:00 AM-5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Saleh Najjar can be reached on 571-272-4006. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Thuong (Tina) Nguyen  
Patent Examiner/Art Unit 2155



SALEH NAJJAR  
SUPERVISORY PATENT EXAMINER